Appeared in:
International Symposium on Information Theory
Seattle, WA; July 2006

Guessing Facets: Polytope Structure and Improved LP Decoder

Alexandros G. Dimakis[1] and Martin J. Wainwright[1,2]
[1] Department of Electrical Engineering and Computer Science
[2] Department of Statistics
University of California, Berkeley
{adim,wainwrig}@eecs.berkeley.edu

***Abstract*— A new approach for decoding binary linear codes by solving a linear program (LP) over a relaxed codeword polytope was recently proposed by Feldman et al. In this paper we investigate the structure of the polytope used in the LP relaxation decoding. We begin by showing that for expander codes, every fractional pseudocodeword always has at least a constant fraction of non-integral bits. We then prove that for expander codes, the active set of any fractional pseudocodeword is smaller by a constant fraction than the active set of any codeword. We exploit this fact to devise a decoding algorithm that provably outperforms the LP decoder for finite blocklengths. It proceeds by guessing facets of the polytope, and resolving the linear program on these facets. While the LP decoder succeeds only if the ML codeword has the highest likelihood over all pseudocodewords, we prove that for expander codes the proposed algorithm succeeds even with a constant number of pseudocodewords of higher likelihood. Moreover, the complexity of the proposed algorithm is only a constant factor larger than that of the LP decoder.**

I. INTRODUCTION

Low-density parity check (LDPC) codes are a class of graphical codes, originally introduced by Gallager [9], that are known to approach capacity as the blocklength increases, even when decoded with the sub-optimal sum-product algorithm. The standard techniques for analyzing the sum-product algorithm, including density evolution [12] and EXIT charts [1], are asymptotic in nature. Many applications, however, require the use of intermediate blocklengths, in which regime asymptotic analysis methods are not suitable for explaining or predicting the behavior of the decoding algorithms. Recently, Feldman et al. [7] introduced the LP decoding method, which is based on solving a linear-programming relaxation of the ML decoder method. While LP decoding performance is not better to message-passing decoders, a possible advantage is its relative amenability to finite-length analysis.

Previous work: The LP decoding idea was introduced by Feldman et al. [4], [7]. There are various theoretical connections between LP decoding and message-passing [3], [10], [15]. For the binary symmetric channel, it can be shown [5] that LP decoding can correct a linear fraction of errors for suitable expander codes. Vontobel and Koetter [14], [10] established bounds on the pseudo-weight for Gaussian channels, showing that it grows only sublinearly for regular codes. Feldman and Stein [6] proved that LP decoding can achieve capacity when applied to generalized expander constructions.

Our contributions: The LP decoder operates by solving a linear program over a polytope $\mathcal{P}$ which constitutes a relaxation of the original combinatorial codeword space. The polytope $\mathcal{P}$ has two types of vertices: *integral vertices* with $0-1$ components corresponding to codewords, and *fractional vertices* that correspond to pseudocodewords. This paper begins by studying the geometric properties of the relaxed polytope. In particular, we prove that for suitable classes of expander codes, the relaxed polytope $\mathcal{P}$ has the property that more facets are adjacent to integral points relative to fractional ones. Motivated by this geometric intuition, we propose an improved LP decoding algorithm that eliminates fractional pseudocodewords by guessing facets of $\mathcal{P}$, and then decodes by re-solving the optimization problem on these facets. We prove some theoretical results on the performance of this facet-guessing decoder. Our experimental results show significant performance improvements, particularly at high SNR, for small and moderate blocklengths.

II. BACKGROUND

Consider a binary linear code with n bits and m checks, and let $R = 1 - \frac{m}{n}$. It can be specified by a parity check matrix $H \in \{0,1\}^{m \times n}$: in particular, the code $\mathbb{C}$ consists of all vectors $x \in \{0,1\}^n$ that satisfy $Hx = 0$, where multiplication and addition are performed over $GF(2)$.

Maximum likelihood decoding as a linear program: The codeword polytope of a code is the convex hull of all its codewords. Maximum likelihood (ML) decoding can be written as a linear program involving the codeword polytope but unfortunately there are no known ways for describing the codeword polytope efficiently. In fact, the existence of a polynomial-time separation oracle for the codeword polytope of a general linear code is very unlikely since ML decoding for arbitrary linear codes is NP-hard [2].

Relaxed polytope and LP decoding: The relaxed polytope $\mathcal{P}$ is an approximation to the codeword polytope that can be described by a linear number of inequalities for LDPC codes. For each check, the corresponding local codeword polytope (LCP) is the convex hull of the bit sequences that satisfy the check (local codewords). For checks of constant bounded degree, the LCP can be described by a constant number of inequalities. The relaxed polytope $\mathcal{P}$ is obtained by looking

at each check independently, and taking the intersection of all the local codeword polytopes.

More specifically, for every check we can find the bit sequences that violate it (local forbidden sequences) and make sure we are sufficiently far away from them. So for every check j connected to variables $N(j)$ find all the possible forbidden sequences S and make sure that their ℓ_1 distance is at least one—viz. $\sum_{N(j)\setminus S} f_i + \sum_{i\in S}(1-f_i) \geq 1$. It can be shown that by picking the ℓ_1 distance to be one we are not excluding any legal codewords from our relaxed polytope. We will call these constraints *forbidden set inequalities*. We also need to add $2n$ inequalities $0 \leq f_i \leq 1$, denoted *box inequality constraints*, which ensure that f remains inside the unit hypercube. It can be shown that for every check, the set of its forbidden inequalities along the box inequalities for the associated variables, describe the LCP of the check. The relaxed polytope is defined as the intersection of all the LCPs (i.e., the constraints consist of all forbidden set inequalities $\mathbb{F}$ along with the box inequalities).

Notice that for every check with degree d_c there is an exponential number of sequences 2^{d_c-1} of local forbidden sequences and therefore the total number of forbidden sequences is $2^{d_c-1}m$. For low-density parity-check codes, d_c is either fixed (for regular) or small with high probability (for irregular) so the number of local forbidden sequences is linear in blocklength. Therefore the relaxed polytope can be described by a linear number of inequalities.

Finally, it can be shown that if the LDPC graph had no cycles, the local forbidden sequences would identify all the possible non-codewords and the relaxation would be exact. However if the graph has cycles, there exist vertices with non $\{0,1\}$ coordinates that satisfy all the local constraints individually and yet are not codewords nor linear combinations of codewords. These sequences are called (fractional) pseudocodewords. To simplify the presentation, we will call all the vertices of the relaxed polytope pseudocodewords (so codewords are also pseudocodewords) and fractional pseudocodewords will be the vertices of the relaxed polytope which happen to have at least one fractional coordinate. One question relates to the number of fractional coordinates (fractional support) that a pseudocodeword can have. While codes can be constructed that have an arbitrarily small fractional support, we show that for expander codes, the fractional support has size at least linear in blocklength. Using this result, we show that for expander codes, the active set of any fractional pseudocodeword (i.e., the number of inequalities that are active at the vertex) is smaller than the active set size of any codeword by at least a linear fraction (in blocklength). These results naturally lead to a randomized algorithm for improving the performance of the LP-decoder by guessing facets of the relaxed polytope and resolving the optimization problem.

III. Structure of the Relaxed Polytope

Definition 1: A (d_c, d_v)-regular bipartite graph is an (α, δ) expander if, for all subsets $|S| \leq \alpha n$, there holds $|N(S)| \geq \delta d_v |S|$.

A. Fractional support of pseudocodewords

A quantity of interest is the fractional support of a pseudocodeword, defined as follows.

Definition 2: The fractional support of a pseudocodeword $x^{\rm pc}$ is the subset $V_{\rm frac}(x^{\rm pc}) \subseteq V$ of bits indices in which $x^{\rm pc}$ has fractional elements. Similarly, the subset of checks that are adjacent to fractional elements of $x^{\rm pc}$ is denoted by $C_{\rm frac}(x^{\rm pc})$.

The following result dictates that all pseudocodewords in an expander code have substantial fractional supports:

Proposition 1: Given an (α, δ)-expander code with $\delta > \frac{1}{2}$, any pseudocodeword has fractional support that grows linearly in blocklength:

$$|V_{\rm frac}(x^{\rm pc})| \;\geq\; \alpha n, \quad \text{and} \quad |C_{\rm frac}(x^{\rm pc})| \;\geq\; \delta d_v \alpha n.$$

Proof: The proof is based on a series of lemmas:

Lemma 1 (Unique neighbor property [13]): Given an (α, δ) expander with $\delta > \frac{1}{2}$, any subset $S \subseteq V$ of size at most αn satisfies the unique neighbor property, i.e there exists $y \in C$ such that $|N(y) \cap S| = 1$.

Proof: Proceed via proof by contradiction: suppose that every $y \in N(S)$ has two or more neighbors in S. Then the total number of edges arriving at $N(S)$ from S is at least $2|N(S)| > 2\delta d_v |S| > d_v |S|$. But the total number of edges leaving S has to be exactly $d_v|S|$, which yields a contradiction. ■

Lemma 2: In any pseudocodeword $x^{\rm pc}$, no check is adjacent to only one fractional variable node.

Proof: Suppose that there exists a check adjacent to only one fractional bit: then the associated local pseudocodeword is in the local codeword polytope (LCP) for this check and therefore can be written as a linear combination of two or more codewords [16]. But these local codewords would have to differ in only one bit, which is not possible for a parity check. ■

We can now prove the main claim. Consider any set S of fractional bits of size $|S| \leq \alpha n$. Using the expansion and Lemma 1, the set $N(S)$ must contain at least one check adjacent to only one bit in S. By Lemma 2, this check must be adjacent to at least one additional fractional bit. We then add this bit to S, and repeat the above argument until $|S| > \alpha n$, to conclude that $|V_{\rm frac}(x^{\rm pc})| > \alpha n$. Finally, the bound on $|C_{\rm frac}(x^{\rm pc})|$ follows by applying the expansion property to a subset of fractional bits of size less than or equal to αn. ■

B. Sizes of active sets

For a vertex v of a polytope, its active set $\mathbb{A}(v)$ is the set of linear inequalities that are satisfied with equality on v. Geometrically, this corresponds to the set of facets of the polytope that contain the vertex v. We want to determine the size of active sets for codewords and pseudocodewords. The key property we want to prove is that for expander codes, codewords have active sets which are larger by at least a constant factor.

Theorem 1: For any (d_v, d_c) code with $R \in (0,1)$, the active set of any codeword $x^{\rm cw}$ has

$$|\mathbb{A}(x^{\rm cw})| \quad = \gamma_{cw} n. \tag{1}$$

elements. For an (α,δ)-expander code with $\delta > \frac{1}{2}$, the active set of any fractional pseudocodeword $x^{\rm pc}$ is smaller than the active set of any codeword by a linear fraction—in particular,

$$|\mathbb{A}(x^{\rm pc})| \quad \leq \quad n\gamma_{pc} \tag{2}$$

where the constants are $\gamma_{cw} = \big[(1-R)d_c + 1\big]$ and $\gamma_{pc} = \Big[\big(1 - R - \delta d_v \alpha\big) d_c + 2\delta d_v \alpha + (1-\alpha)\Big]$. (Note that $\gamma_{pc} < \gamma_{cw}$.)

Proof: We begin by proving equation (1). By the code-symmetry of the relaxed polytope [7], every codeword has the same number of active inequalities, so it suffices to restrict our attention to the all-zeroes codeword. The check inequalities active at the all-zeros codeword are in one-to-one correspondence with those forbidden sequences at Hamming distance 1. Note that there are d_c such forbidden sequences, so that the total number of constraints active at the all-zeroes codeword is simply $|\mathbb{A}(x^{\rm cw})| = md_c + n \;=\; n\left[(1-R)d_c + 1\right]$ as claimed.

We now turn to the proof of the bound (2) on the size of the fractional pseudocodeword active set. Recall that the relaxed polytope consists of two types of inequalities: *forbidden set constraints* (denoted $\mathbb{F}$) associated with the checks, and the *box inequality constraints* $0 \leq x_i \leq 1$ (denoted $\mathbb{B}$) associated with the bits. The first ingredient in our argument is the fact (see Proposition 1) that for an (α,δ)-expander, the fractional support $V_{\rm frac}(x^{\rm pc})$ is large, so that a constant fraction of the box inequalities will not be active.

Our second requirement is a bound on the number of forbidden set inequalities that can be active at a pseudocodeword. We establish a rough bound for this quantity using the following lemma:

Lemma 3: Suppose that z belongs to a polytope and is not a vertex. Then there always exist at least two vertices x, y such that $\mathbb{A}(z) \subseteq \mathbb{A}(x) \cap \mathbb{A}(y)$.

Proof: Since z belongs to the polytope but is not a vertex, it must either belong to the interior, or lie on a face with dimension at least one. If it lies in the interior, then $\mathbb{A}(z) = \emptyset$, and the claim follows immediately. Otherwise, z must belong to a face F with $\dim(F) \geq 1$. Then F must contain [16] at least $\dim(F) + 1 = 2$ vertices, say x and y. Consequently, since x, y and z all belong to F and z is not a vertex, we must have $\mathbb{A}(z) \subseteq \mathbb{A}(y)$ and $\mathbb{A}(z) \subseteq \mathbb{A}(x)$, which yields the claim. ■

Given a check c and codeword $x^{\rm cw}$, let $\Pi_c(x^{\rm cw})$ denote the restriction of $x^{\rm cw}$ to bits in the neighborhood of c (i.e., a *local codeword* for the check c). With this notation, we have:

Lemma 4: For any two local codewords $\Pi_c(x_1^{\rm cw})$ and $\Pi_c(x_2^{\rm cw})$ of a check c, the following inequality holds

$$|\mathbb{A}(\Pi_c(x_1^{\rm cw})) \cap \mathbb{A}(\Pi_c(x_2^{\rm cw}))| \leq 2. \tag{3}$$

Proof: The intersection $\mathbb{A}(\Pi_c(x_1^{\rm cw})) \cap \mathbb{A}(\Pi_c(x_2^{\rm cw}))$ is given by the forbidden sequences that have Hamming distance 1 from $\Pi_c(x_i^{\rm cw}), i = 1, 2$ (i.e., forbidden sequences f such that $d(f, \Pi_c(x_i^{\rm cw})) = 1$ for $i = 1, 2$). Thus, if such an f exists, then by the triangle inequality for Hamming distance, we have

$$2 = d(f, \Pi_c(x_1^{\rm cw})) + d(f, \Pi_c(x_2^{\rm cw}))) \geq d(\Pi_c(x_1^{\rm cw}), \Pi_c(x_2^{\rm cw})), \tag{4}$$

But $d(\Pi_c(x_1^{\rm cw}), \Pi_c(x_2^{\rm cw})) \geq 2$ for any two local codewords, so that we must have $d(\Pi_c(x_1^{\rm cw}), \Pi_c(x_2^{\rm cw})) = 2$. Consequently, we are looking for all the forbidden (odd) sequences of length d_c that differ in one bit from two local codewords that are different in two places. Clearly there are only two such forbidden sequences, so that the claim follows. ■

We can now establish a bound on the size of the active sets of pseudocodewords for (α,δ)-expanders:

Lemma 5: For every pseudocodeword $x^{\rm pc}$, the size of the active set $|\mathbb{A}(x^{\rm pc})|$ is upper bounded by

$$(m - |C_{\rm frac}(x^{\rm pc})|)d_c + 2|C_{\rm frac}(x^{\rm pc})| + n - |V_{\rm frac}(x^{\rm pc})|. \tag{5}$$

Proof: The proof is based on the decomposition:

$$|\mathbb{A}(x^{\rm pc})| \quad = \quad |\mathbb{A}(x^{\rm pc}) \cap \mathbb{F}| + \mathbb{A}(x^{\rm pc})| \cap \mathbb{B}|.$$

The cardinality $|\mathbb{A}(x^{\rm pc}) \cap \mathbb{B}|$ is equal to the number of integral bits in the pseudocodeword, given by $n - |V_{\rm frac}(x^{\rm pc})|$.

We now turn to upper bounding the cardinality $|\mathbb{A}(x^{\rm pc}) \cap \mathbb{F}|$. Consider the $m - |C_{\rm frac}(x^{\rm pc})|$ checks that are adjacent to only integral bits of $x^{\rm pc}$. For each such check, exactly d_c forbidden set constraints are active, thereby contributing a total of $d_c\big[m - |C_{\rm frac}(x^{\rm pc})|\big]$ active constraints. Now consider one of the remaining $|C_{\rm frac}(x^{\rm pc})|$ fractional checks, say c. Consider the restriction $\Pi_c(x^{\rm pc})$ of the pseudocodeword $x^{\rm pc}$ to the check neighborhood of c. Since $\Pi_c(x^{\rm pc})$ contains fractional elements, it is not a vertex of the local codeword polytope associated with c. Therefore, by combining Lemmas 3 and 4, we conclude that $|\mathbb{A}(\Pi_c(x^{\rm pc}))| \leq 2$. Overall, we conclude that the upper bound (5) holds. ■

Using Lemma 5 and Proposition 1, we can now complete the proof of Theorem 1. In particular, we re-write the RHS of the bound (5) as $(1-R)d_c\, n - (d_c - 2)|C_{\rm frac}(x^{\rm pc})| + n - |V_{\rm frac}(x^{\rm pc})|$. From Proposition 1, we have $|C_{\rm frac}(x^{\rm pc})| \geq d_v \delta \alpha n$ and $|V_{\rm frac}(x^{\rm pc})| > \alpha n$, from which the bound (2) follows. ■

IV. Improved LP decoding

Various improvements to the standard sum-product decoding algorithm have been suggested in past work [e.g., 8], [11]. Based on the structural results that we have obtained, we now describe some improved decoding algorithms for which some finite-length analysis is possible. We begin with some simple observations: (i) ML decoding corresponds to finding the vertex in the relaxed polytope that has the highest likelihood and integral coordinates; and (ii) Standard LP decoding succeeds if and only if the ML codeword has the highest likelihood over all pseudocodewords.

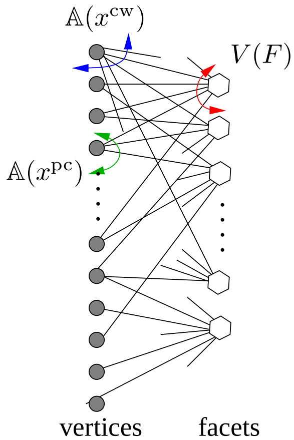


Fig. 1. Vertex-facet diagram of the relaxed polytope. Circles on the left-hand side correspond to vertices (codewords x^{cw} and fractional pseudocodewords x^{pc}) of the relaxed polytope; hexagons on the right-hand side correspond to facets (hyperplane inequalities) defining the relaxed polytope.

These observations highlight the distinction between LP decoding and ML decoding. An LP solver, given the (polynomially many) facets of the relaxed polytope, determines the vertex with the highest likelihood without having to go through all the exponentially many vertices of V. In contrast, the ML decoder can go down this list, and determine the first vertex which has integral coordinates. This motivates facet-guessing: suppose that there exists only one fractional pseudocodeword x_1^{pc} that has higher likelihood than the ML codeword x^{cw}. The LP decoder will output the pseudocodeword x_1^{pc}, resulting in a decoding error. However, now suppose that there exists a facet $F_1 \in \mathbb{A}$ such that $x^{\mathrm{cw}} \in F_1$ but $x^{\mathrm{pc}} \notin F_1$. Consider the reduced polytope $\mathcal{P}'$ created by restricting the relaxed polytope $\mathcal{P}$ to the facet F_1 (i.e., $\mathcal{P}' = \mathcal{P} \cap F_1$). This new polytope will have a vertex-facet graph $\mathcal{B}'$ with vertices $V' = N(F_1)$ i.e. all the vertices that are contained in F_1. The likelihoods will be the same, but p_1 will not belong in $\mathcal{P}'$ and therefore we can use an LP solver to determine the vertex with the highest likelihood in $\mathcal{P}'$ which will be c. Therefore if we could guess the right facet F_1 we can determine the ML codeword for this case. Based on this intuition, we introduce two postprocessing algorithms for improving LP decoding.

Facet Guessing Algorithm

1) Run LP decoding: if outputs an integral codeword, terminate. Otherwise go to Step 2.
2) Take as input:
   - fractional pseudocodeword x^{pc} from the LP decoder
   - likelihood vector γ.
3) Given a natural number$N \geq 1$, repeat for $i = 1, \ldots N$:
   (a) Select a facet $F_i \in (\mathbb{A} \setminus \mathbb{A}_{x^{\mathrm{pc}}})$, form the reduced polytope a new polytope $\mathcal{P}' = \mathcal{P} \cap F_i$.
   (b) Solve the linear program with objective vector γ in $\mathcal{P}'$, and save the optimal vertex z_i.
4) From the list of optimal LP solutions $\{z_1, \ldots, z_N\}$, output the integral codeword with highest likelihood.

Remarks: (a) There are two variations of facet guessing: exhaustive facet guessing (EFG) tries all possible facets (i.e., $N = |(\mathbb{A} \setminus \mathbb{A}_{x^{\mathrm{pc}}})|$), while randomized facet guessing (RFG) randomly samples from $(\mathbb{A} \setminus \mathbb{A}_{p_i})$ a constant number of times (e.g., $N = 20$). (b) Note that the EFG algorithm has polynomial-time complexity. Since $|\mathbb{A} \setminus \mathbb{A}_{x^{\mathrm{pc}}}| = O(n)$ this requires only a linear number of calls to an LP solver. On the other hand, the RFG algorithm requires a constant number of calls to an LP solver and therefore has the same complexity order as LP decoding. We now provide a characterization of when the EFG algorithm fails:

Lemma 6: The exhaustive facet-guessing algorithm fails to find the ML codeword $c \iff$ every facet $F \in \mathbb{A}_c$ contains a fractional pseudocodeword with likelihood greater than c.

Proof: Denote the set of fractional pseudocodewords with likelihood higher than c by $\hat{p}$. Assume there exists a facet F_i such that $c \in F_i$ and $\forall p \in \hat{p},\ p \notin F_i$. Then the algorithm will at some point select F_i and the LP solver will output the vertex in $\mathcal{P}'$ with the highest likelihood which will be c since nothing from $\hat{p}$ can belong in $\mathcal{P}'$. Therefore c will be in the list of LP solutions. Also, since c is the ML codeword, there can be no other integral codeword with higher likelihood in the list, and therefore the algorithm will output c. ■

By using this characterization and Theorem 1 for expander codes, we obtain the following result:

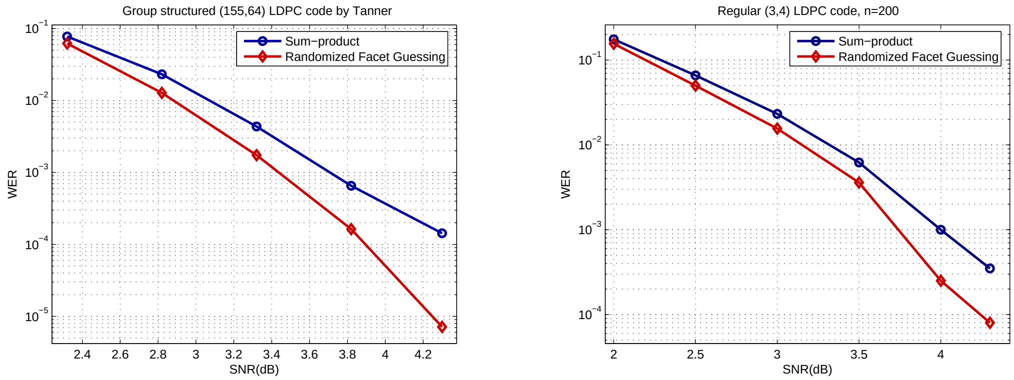
Corollary 1: For expander codes, the EFG algorithm will always succeed if there are C_1 fractional pseudocodewords with likelihood higher than the ML codeword and $C_1 < \frac{\gamma_{cw}}{\gamma_{pc}}$. Under this condition, each iteration of RFG succeeds with constant probability $p_{RFG} \geq \frac{\gamma_{cw} - C_1\gamma_{pc}}{2^{d_c-1}(1-R)+2}$.

Proof: From Lemma 6, the EFG algorithm fails if and only if every facet in $|\mathbb{A}_c|$ also contains another fractional pseudocodeword with higher likelihood. But for expander codes, Lemma 5 yields that the size of the active set of any fractional pseudocodeword is upper bounded as

$$|\mathbb{A}_p| \quad \leq n\gamma_{pc}.$$

while the size of active sets of any codeword is always $|\mathbb{A}_c| = n\gamma_{cw}$. Therefore, if there exist C_1 fractional pseudocodewords with likelihood higher than c, the total number of facets adjacent to these fractional pseudocodewords is at most $\gamma_{pc}C_1 n$. Therefore when $\gamma_{pc}C_1 n < n\gamma_{cw}$ it is impossible to completely cover $\mathbb{A}_c$ and EFG succeeds. Also RFG at each iteration selects a random facet and there are $(\gamma_{cw} - \gamma_{pc}C_1)n$ facets that contain c but not any fractional pseudocodeword with higher likelihood. The total number of facets is $|\mathbb{A}| = (2^{d_c-1}(1-R)+2)n$ and therefore each iteration of RFG has probability of success larger than $\frac{\gamma_{cw} - C_1\gamma_{pc}}{2^{d_c-1}(1-R)+2}$. ■

Notice that this corollary only provides a worst case bound. Even though there is a linear number of facets that contain the ML codeword, we show that it will require a constant number of fractional pseudocodewords to cover them. This can only happen if the high likelihood fractional pseudocodewords have their adjacent facets non-overlapping and entirely



(a) (b)

Fig. 2. Comparison of different decoding methods: standard sum-product decoding, and randomized facet-guessing (RFG) with $N = 20$ iterations. The two panels show two different codes: (a) Tanner's group-structured code. (b) Random (3,4) LDPC code with $n = 200$.

contained in A_c. More typically, one could expect the facet guessing algorithm to work even if there are many more fractional pseudocodewords with higher likelihoods. Indeed, our experimental results show that the RFG algorithm leads to a significant performance gain for those codewords that are recovered successfully by neither sum-product nor LP decoding. As shown in Figure 2, the gains are pronounced for higher SNR, as high as 0.5dB for the small blocklengths that we experimentally tested. The added complexity corresponds to solving a constant number of LP optimizations; moreover, the extra complexity is required *only if* LP decoding fails.

V. Discussion

We have investigated the structure of the polytope that underlies both LP decoding and the sum-product algorithm. We show that for expander codes, every fractional pseudocodeword always has at least a constant fraction of non-integral bits. We further proposed an decoding algorithm, with complexity only a constant factor larger than that of the LP decoder, and analyzed the performance gains that it achieves. This theoretical analysis is supplemented with experimental results showing gains for short to moderate block lengths, particularly at high SNR.

Acknowledgment

Work partially supported by NSF Grant DMS-0528488, and a UC-Micro grant through Marvell Semiconductor.

References

[1] A. Ashihkmin, G. Kramer, and S. ten Brink. Extrinsic information transfer functions: model and erasure channel properties. *IEEE Trans. Info. Theory*, 50(11):2657–2673, 2004.

[2] E. Berlekamp, R. McEliece, and H. van Tilborg. On the inherent intractability of certain coding problems. *IEEE Trans. Info. Theory*, pages 384–386, 1978.

[3] J. Feldman, D. R. Karger, and M. J. Wainwright. Linear programming-based decoding of turbo-like codes and its relation to iterative approaches. In *Proc. 40th Annual Allerton Conf. on Communication, Control, and Computing*, October 2002.

[4] J. Feldman, D. R. Karger, and M. J. Wainwright. Using linear programming to decode LDPC codes. In *Conference on Information Science and Systems*, March 2003.

[5] J. Feldman, T. Malkin, R. A. Servedio, C. Stein, and M. J. Wainwright. LP decoding corrects a constant fraction of errors. Technical Report CORC Technical Report TR-2003-08, Operations Research, Columbia University, December 2003.

[6] J. Feldman and C. Stein. LP decoding achieves capacity. In *SODA*, January 2005.

[7] J. Feldman, M. J. Wainwright, and D. R. Karger. Using linear programming to decode binary linear codes. *IEEE Transactions on Information Theory*, 51:954–972, March 2005.

[8] M. P. C. Fossorier. Iterative reliability-based decoding of low-density parity check codes. *IEEE Transactions on Information Theory*, pages 908–917, May 2001.

[9] R. G. Gallager. *Low-density parity check codes*. MIT Press, Cambridge, MA, 1963.

[10] R. Koetter and P. O. Vontobel. Graph-covers and iterative decoding of finite length codes. In *Proc. 3rd International Symp. on Turbo Codes*, September 2003.

[11] H. Pishro-Nik and F. Fekri. On decoding of LDPC codes over the erasure channel. *IEEE Trans. Inform. Theory*, 50:439–454, 2004.

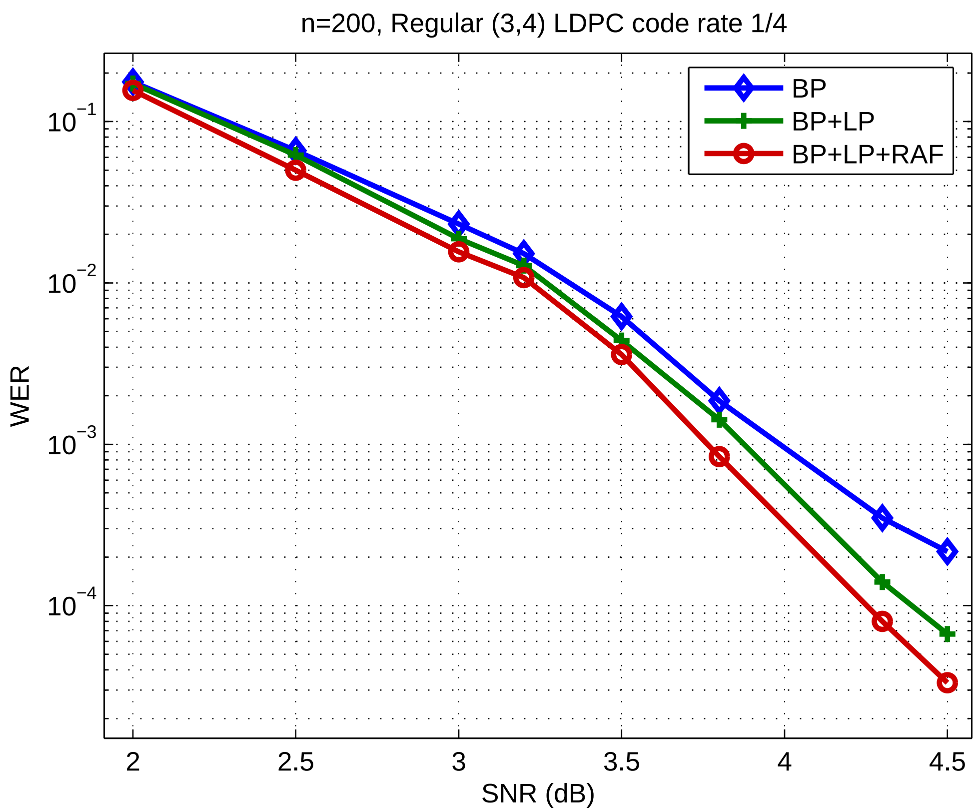
[12] T. Richardson and R. Urbanke. The capacity of low-density parity check codes under message-passing decoding. *IEEE Trans. Info. Theory*, 47:599–618, February 2001.

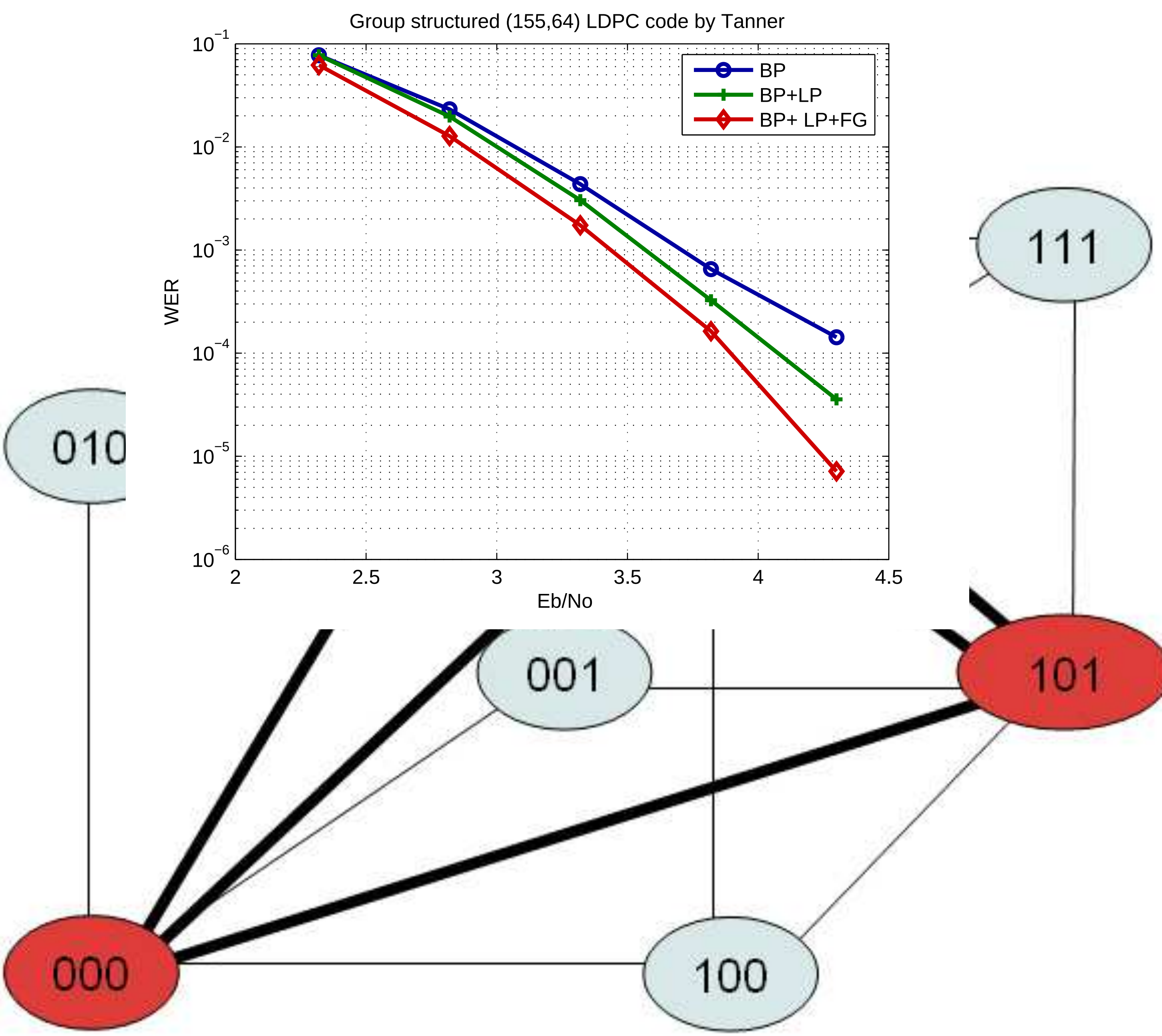
[13] M. Sipser and D. Spielman. Expander codes. *IEEE Trans. Info. Theory*, 42:1710–1722, November 1996.

[14] P. Vontobel and R. Koetter. Lower bounds on the minimum pseudo-weight of linear codes. In *International Symposium on Information Theory (ISIT '04), Chicago, IL*, June 2004.

[15] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky. Exact MAP estimates via agreement on (hyper)trees: Linear programming and message-passing. *IEEE Trans. Information Theory*, 51(11):3697–3717, November 2005.

[16] G. M. Ziegler. *Lectures on polytopes*. Springer-Verlag, New York, 1995.


n=200, Regular (3,4) LDPC code rate 1/4
BP
BP+LP
BP+LP+RAF
WER
10^{-1}
10^{-2}
10^{-3}
10^{-4}
2
2.5
3
3.5
4
4.5
SNR (dB)


Group structured (155,64) LDPC code by Tanner
WER
Eb/No
BP
BP+LP
BP+ LP+FG
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
2
2.5
3
3.5
4
4.5
111
001
101
000
100